Fenwick Tree

(binary indexed tree)

The Problem

- There are several boxes
  - Labeled from 1 to N
- We can
  - Add N marble(s) into i^{th} box
    - We say box #i has frequency N
- We want to know
  - Total number of marbles in box #1 to #j

Fenwick Tree

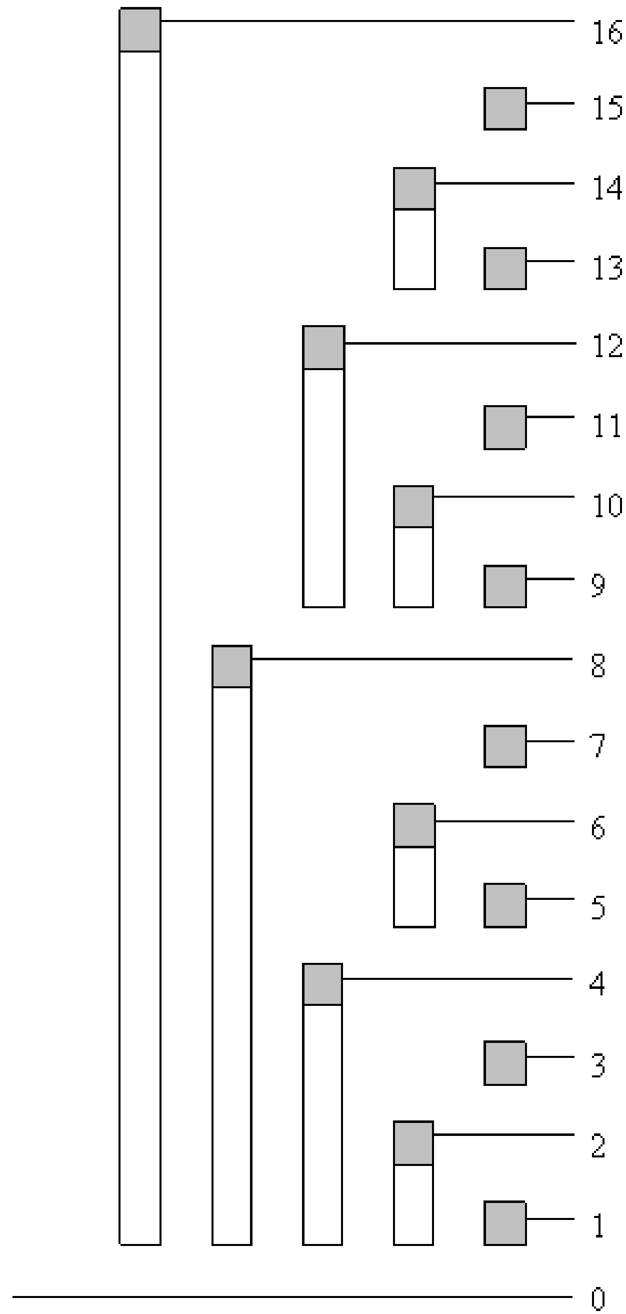
- Operation
  - `void create(int n);` O(N)
  - `void update(int idx, int val);` O(log N)
  - `int freqTo(int idx);` O(log N)
  - `int freqAt(int idx);` O(log N)

Storage

- Data
  - An int array of size N

Fenwick Tree

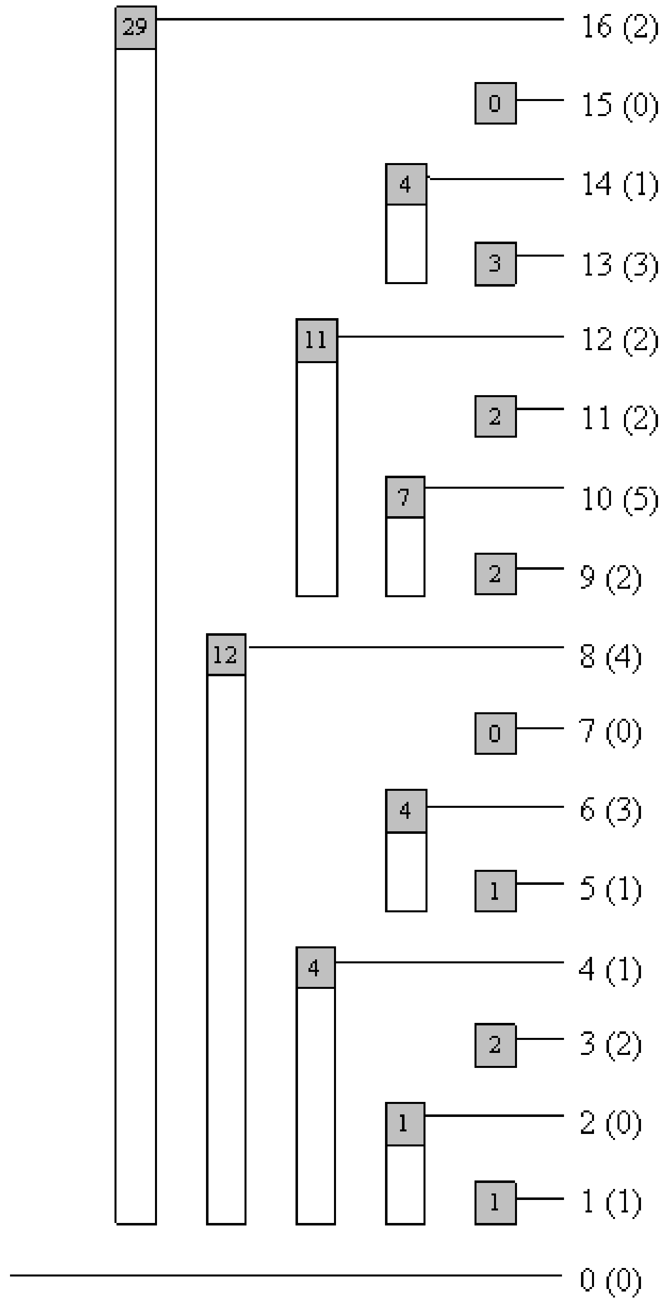
- How it works?
  - Each element in the array stores cumulative frequency of consecutive list of boxes
  - Range of boxes that is stored is related to “binary value” of the index



pic from www.topcoder.com

Define

- f(x) = number of marble in box x
- c(x) = summation of number of marble in box #1 to box #x
- tree[x] = element x in the array



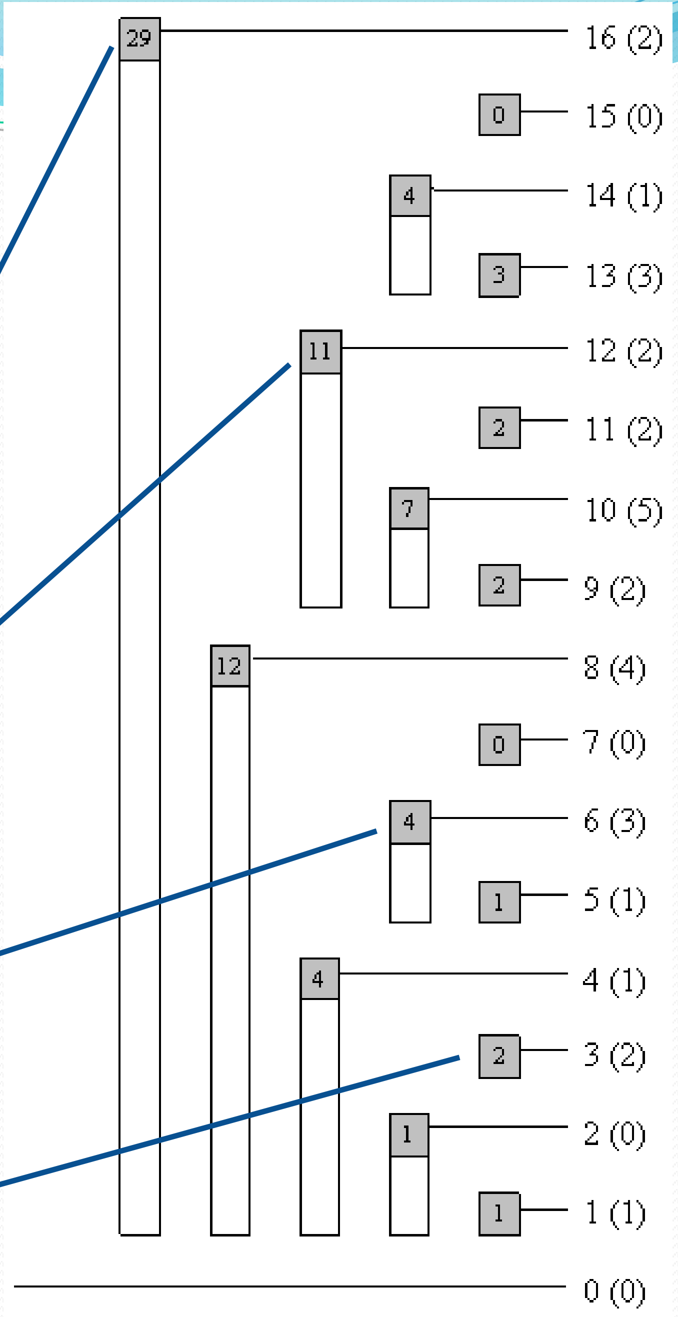
Storage Solution

Tree[16] = f(1) + f(2) + … + f(16)

Tree[12] = f(9) + f(10) + … + f(12)

Tree[6] = f(5) + f(6)

Tree[3] = f(3)



f(16) = 2

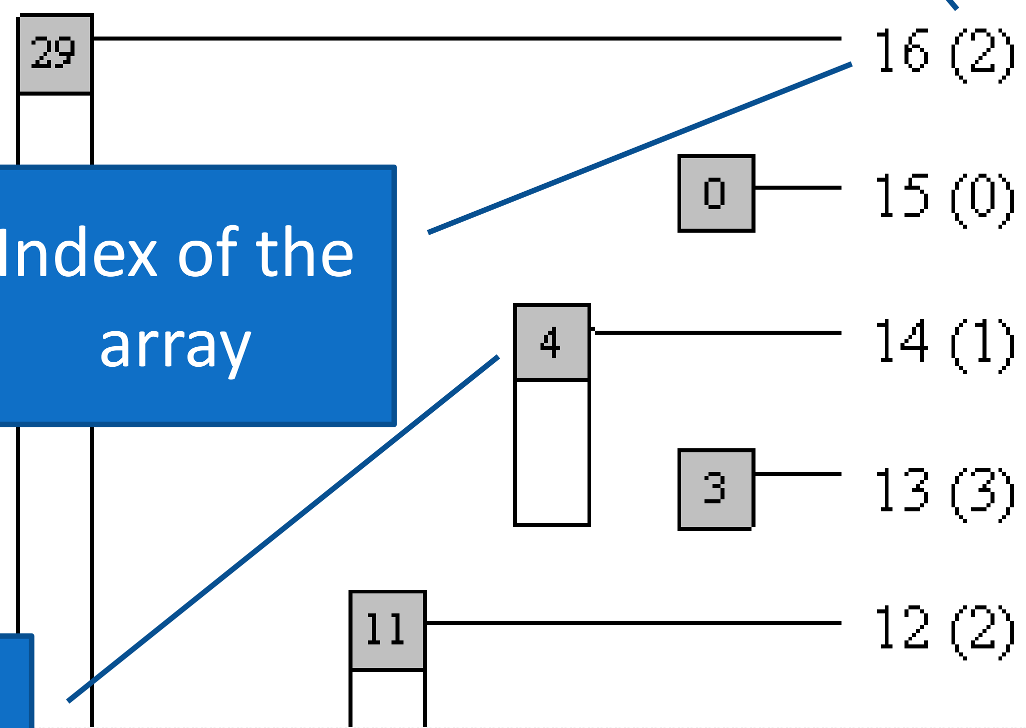
Cumulative Freq

Actual frequency

tree[16] = 29

Cumulative frequency From 1 to 16

Index of the array



tree[14]

Cumulative frequency From 13 to 14

pic from www.topcoder.com

The last 1

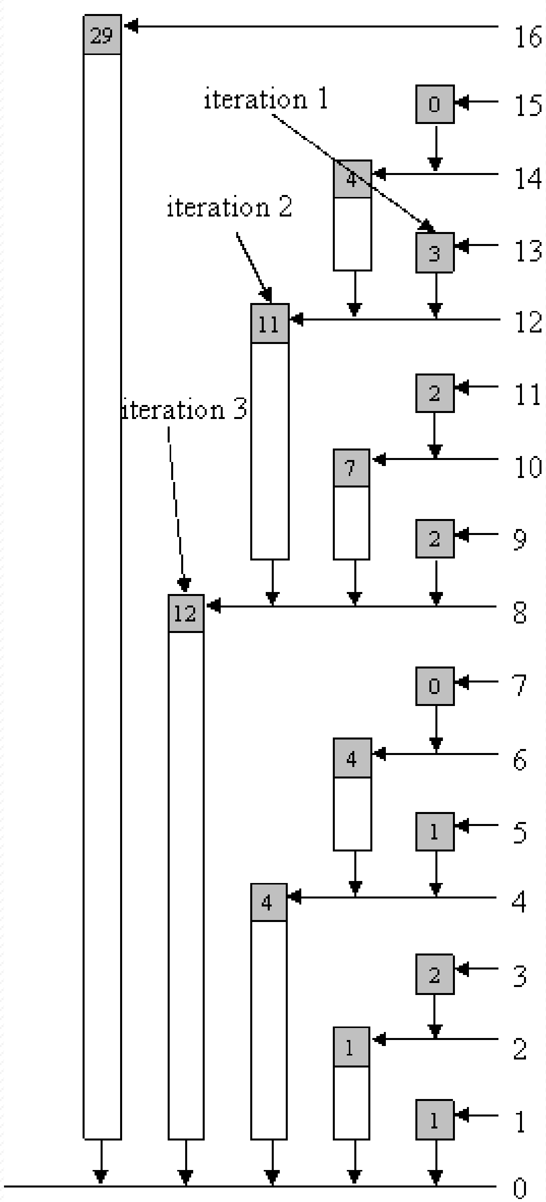
- A node at the index X will store freq of boxes in the range
  - $X - 2^r + 1$ to X
  - Where r is the position of the last digit of 1

- Ex
  - $X = 12\ (1100)_2$
  - Node will store freq from 9 to 12
    - The last 1 of 12 is at position 2 (0-indexed)
    - $12 - 2^2 + 1 = 9 = (1001)_2$

Read Cumulative Freq

c(13) =
tree[13] +
tree[12] +
tree[8]

In base-2
$c(1101_2)$ =
$tree[1101_2]$ +
$tree[1100_2]$ +
$tree[1000_2]$



pic from www.topcoder.com

Update Freq

Update f(5) by -1

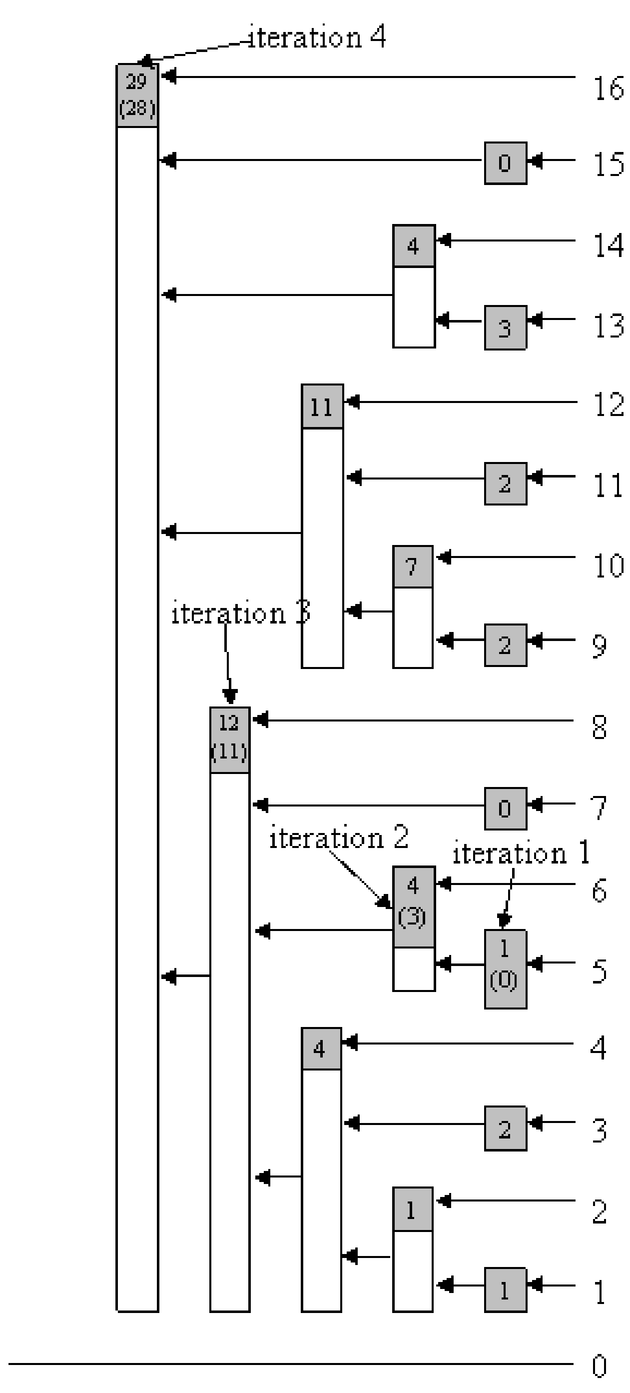
involve

Tree[16] (10000_2)

Tree[8] (01000_2)

Tree[6] (00110_2)

Tree[5] (00101_2)



pic from www.topcoder.com

Read actual Freq

What is f(12)?

Easy, it's c(12) – c(11)

easier

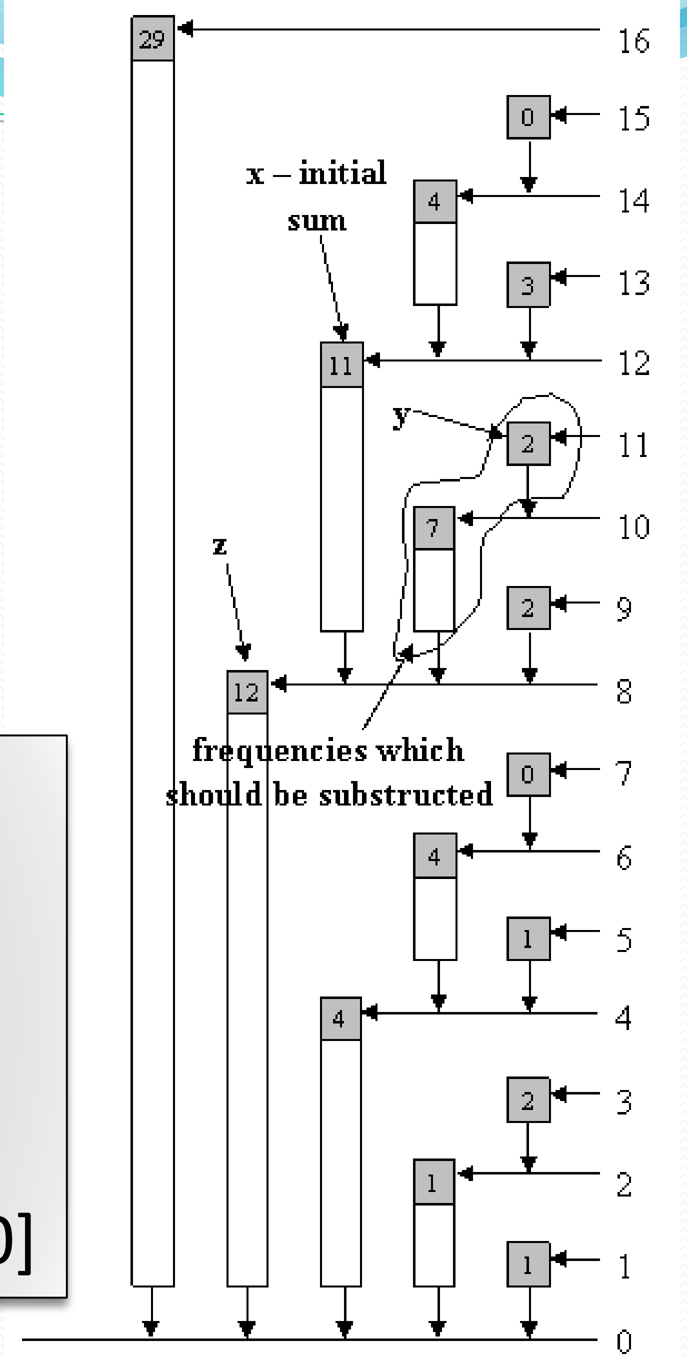
Tree[12] = f(9) + f(10) + f(11) + f(12)

Tree[11] = f(11)

Tree[10] = f(9) + f(10)

Hence,

f(12) = Tree[12] – Tree[11] – Tree[10]



pic from www.topcoder.com

Two’s compliment

- A method to represent negative
  - A two’s compliment of X is
    - (compliment of x) + 1
  - Ex.. 2’s Compliment of 7 is
    - 0111 ➔ 1000 ➔ 1001

- Finding the last 1
- x = a1b
  - b = consecutive of 0
- Ex... X = 4 = 0100
  - a = 0 b = 00

0111	**7**
0110	**6**
0101	**5**
0100	**4**
0011	**3**
0010	**2**
0001	**1**
0000	**0**
1111	**−1**
1110	**−2**
1101	**−3**
1100	**−4**
1011	**−5**
1010	**−6**
1001	**−7**
1000	**−8**

Two's compliment

- Now, let's see two's compliment more closely
- -x
  - $= \overline{(a1b)} + 1$
  - $= \bar{a}0\bar{b} + 1$
  - $= \bar{a}0\overline{(0...0)} + 1$
  - $= \bar{a}0(1...1) + 1$
  - $= \bar{a}1(0...0)$
  - $= \bar{a}1b$.
- So, if we "&" –x and x
  - $\bar{a}1b$ & $a1b$.
  - We got the last 1

0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Code

```
int freqTo(int idx) {
        int sum = 0;
        while (idx > 0){
                sum += tree[idx];
                idx -= (idx & -idx);
        }
        return sum;
}
```

```
void update(int idx ,int val) {
     while (idx <= MaxVal){
          tree[idx] += val;
          idx += (idx & -idx);
     }
}
```

from www.topcoder.com

Code

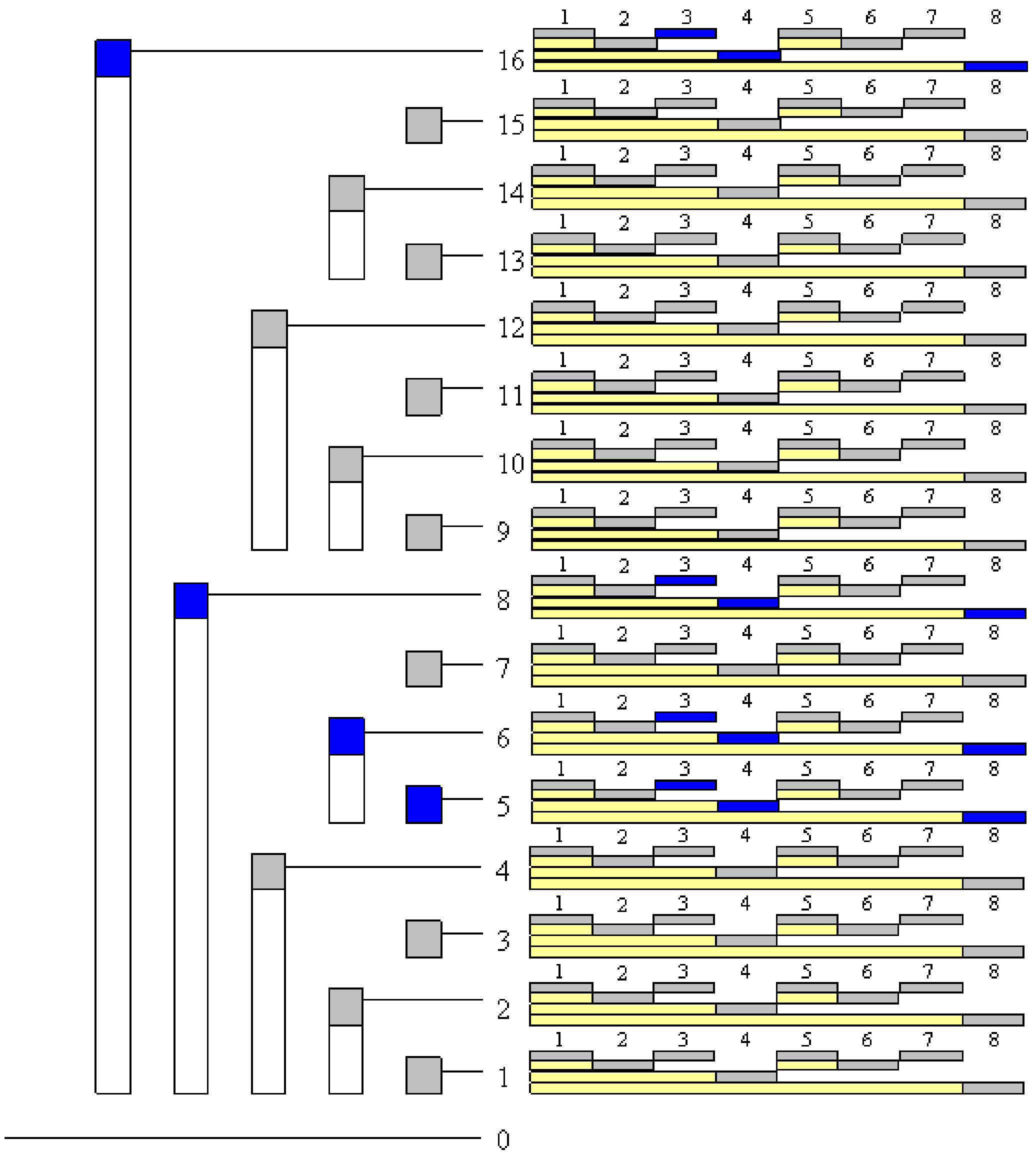
```
int freqAt(int idx){
        int sum = tree[idx];
        if (idx > 0) {
                int z = idx - (idx & -idx);
                y = idx - 1;
                while (y != z){
                        sum -= tree[y];
                        y -= (y & -y);
                }
        }
        return sum;
}
```

from www.topcoder.com

2D BIT

- Box is arrange at x-y coordinate
- Operation
  - Update(x,y,val)   (add “val” marble in position (x,y))
  - How many points in the range (x1,y1) to (x2,y2)

2D BIT



pic from www.topcoder.com